Training robust neural networks

DataLab — Assignment 3

Alexandre Vérine

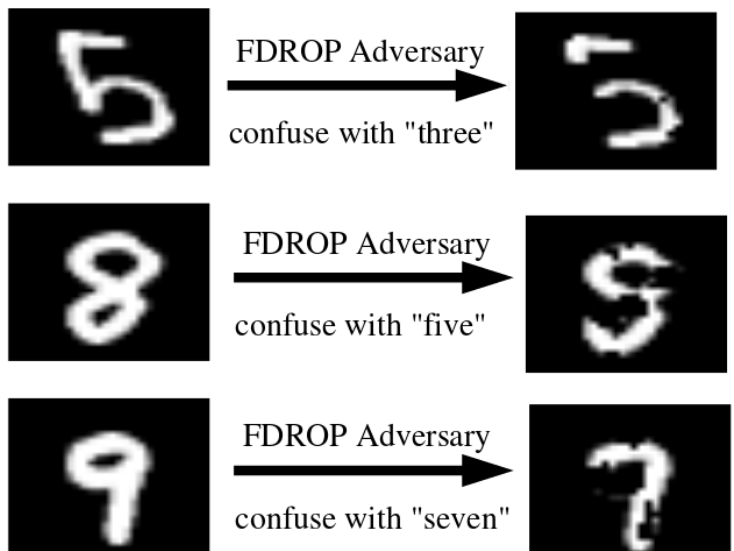
Université Paris Dauphine – PSL / ENS – PSL

12 November 2026



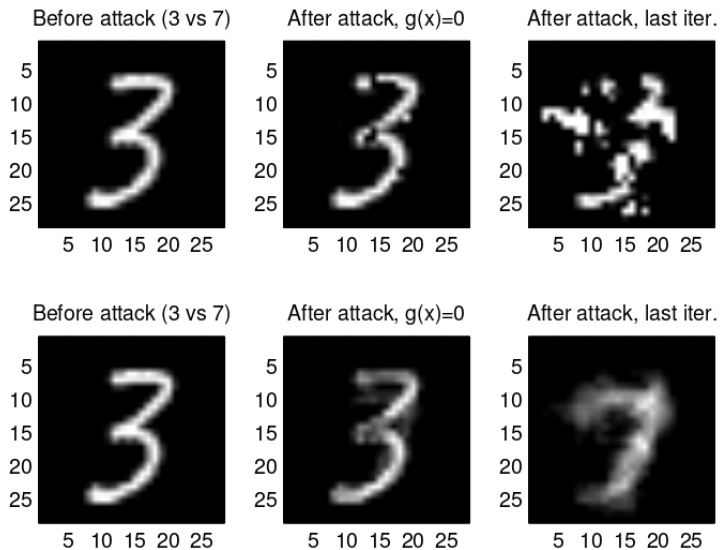
Adversarial examples

Early work: feature deletion



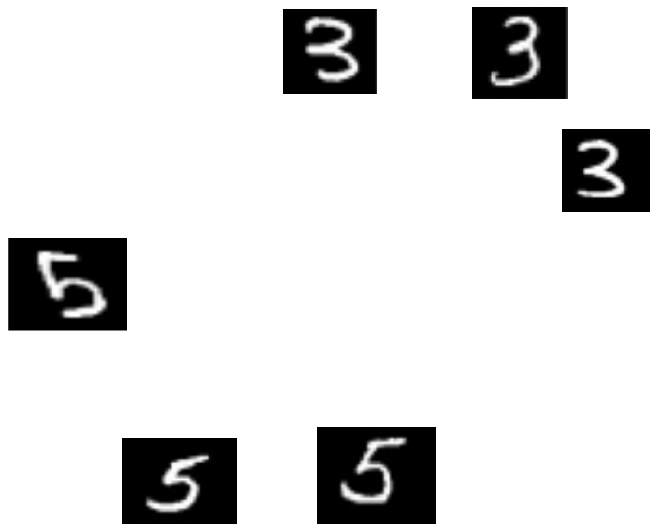
Globerson & Roweis · ICML 2006

Early work: adversarial examples at test time

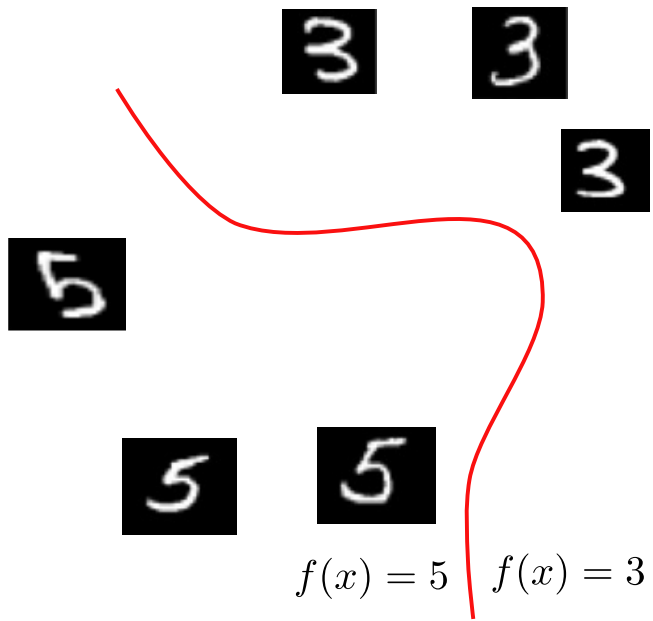


Biggio et al. · ECML PKDD 2013

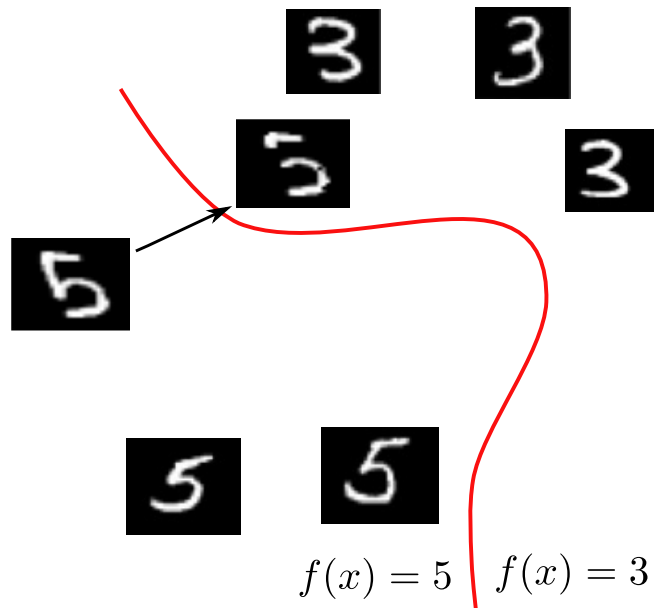
Adversarial examples explained



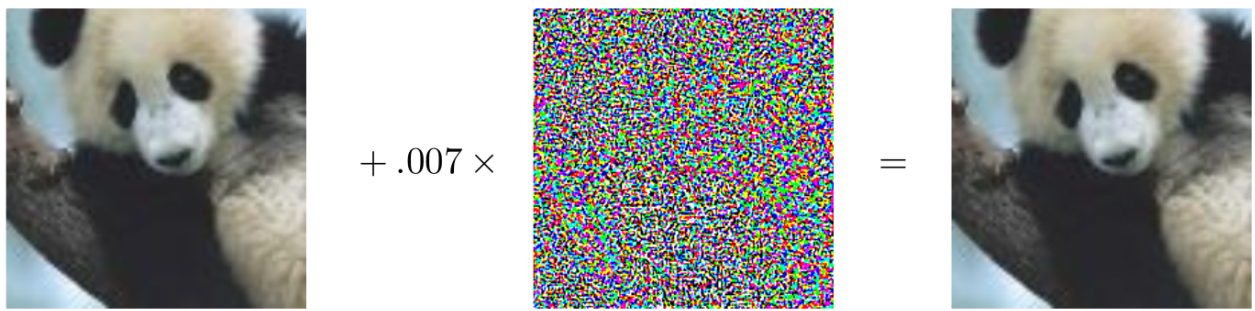
Adversarial examples explained



Adversarial examples explained



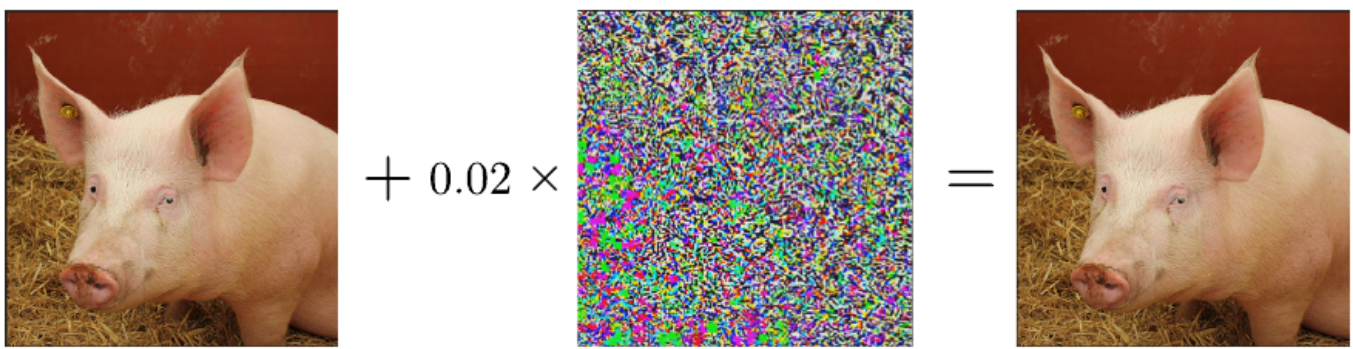
A small change, a different prediction



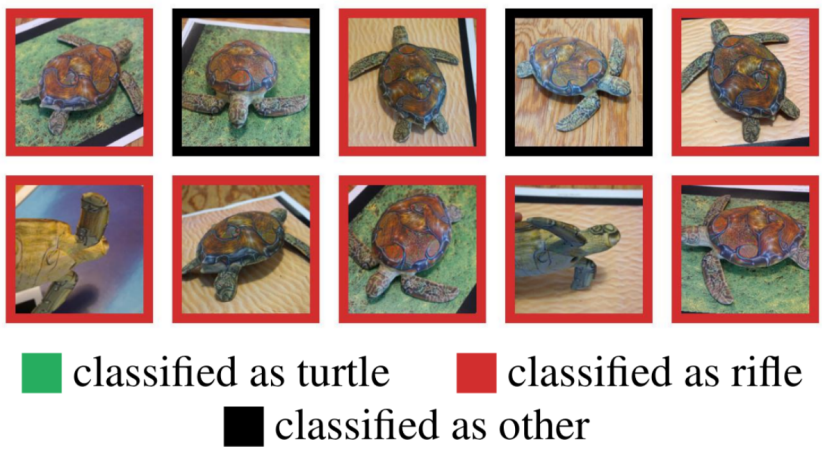
Goodfellow, Shlens & Szegedy · ICLR 2015 Explaining and Harnessing Adversarial Examples

The perturbation is chosen to change the model’s decision, rather than sampled at random.

Pig versus airliner



Adversarial examples can survive physical transformations

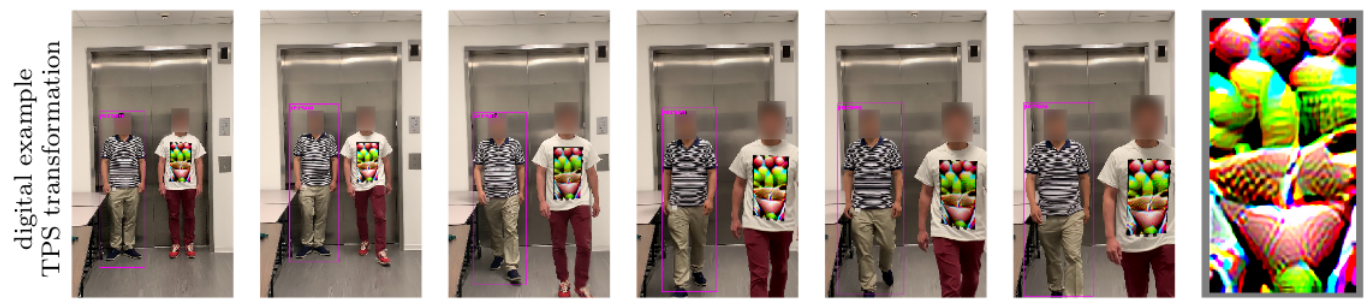


Athalye, Engstrom, Ilyas & Kwok · ICML 2018 Synthesizing Robust Adversarial Examples

Changing viewpoint, lighting or camera noise makes the attack problem harder.

An attack can optimise over a distribution of transformations.

Adversarial clothing



Xu et al. (2019 preprint), *Evading Real-Time Person Detectors by Adversarial T-shirt*.

What will you investigate?

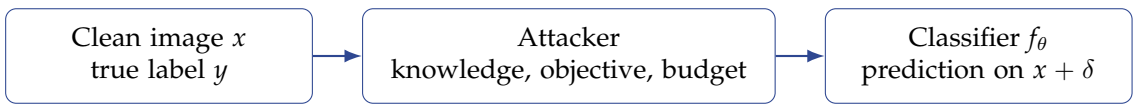
Attack
How can a small, deliberately chosen perturbation change a prediction?

Defence
How can training or prediction make that change harder?

Evidence
Does the improvement survive a stronger evaluation?

Interpretation
Which mechanism explains the result, and where does it fail?

Define the threat model before comparing methods



- What may change? Pixels, with $x + \delta \in [0, 1]^d$ in this example.
- How much? A specified norm and perturbation budget ε.
- What does the attacker know? Weights and gradients in the white-box setting.

Generating adversarial examples

Let $f : \mathbb{R}^d \to \{1, \ldots, C\}$ be a classifier.

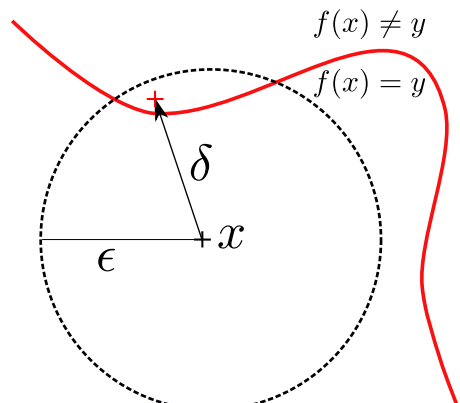
Untargeted

Find δ such that $f(x + \delta) \neq y$.

Targeted

Choose $t \neq y$ and require $f(x + \delta) = t$.

In both cases: $\|\delta\|_p \leq \varepsilon$ and $x + \delta \in [0, 1]^d$.



Generating adversarial examples

Let $f : \mathbb{R}^d \to \{1, \dots, C\}$ be a classifier.

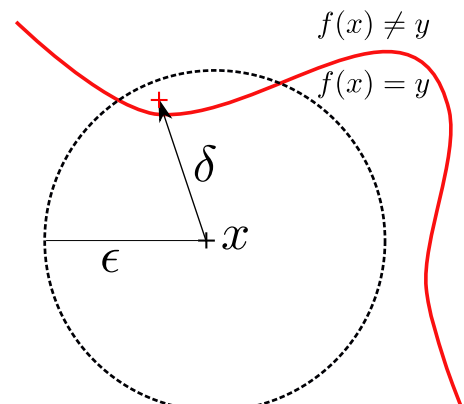
Untargeted

Find δ such that $f(x + \delta) \neq y$.

Targeted

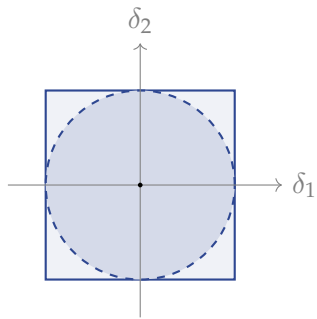
Choose $t \neq y$ and require $f(x + \delta) = t$.

In both cases: $\|\delta\|_p \leq \varepsilon$ and $x + \delta \in [0, 1]^d$.



$$\delta^* \in \underset{\substack{\|\delta\|_p \leq \varepsilon \\ x+\delta \in [0,1]^d}}{\operatorname{argmax}} \ell\big(f_\theta(x + \delta), y\big).$$

Two perturbation budgets, two different neighbourhoods



Equal radii, different admissible sets

$$\|\delta\|_2 = \sqrt{\sum_i \delta_i^2} \leq \varepsilon_2$$

Total perturbation energy.

$$\|\delta\|_\infty = \max_i |\delta_i| \leq \varepsilon_\infty$$

Maximum change to any component.

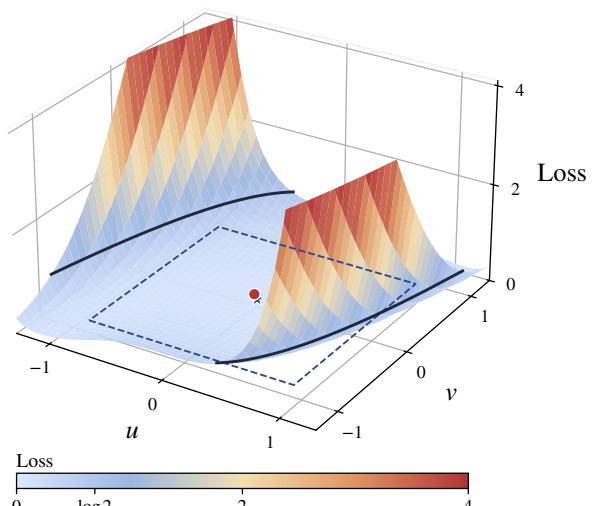
A norm constraint is a mathematical model of similarity, not a guarantee of perceptual similarity.

Build and evaluate attacks

THE ATTACK MOVES THE INPUT UPHILL IN LOSS

Increase the loss to change the prediction

L-infinity steps use the gradient signs; the model stays fixed.



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Climb the same loss surface

Take small signed-gradient steps.
Same start as the FGSM example.
Here, the attack reaches its corner.

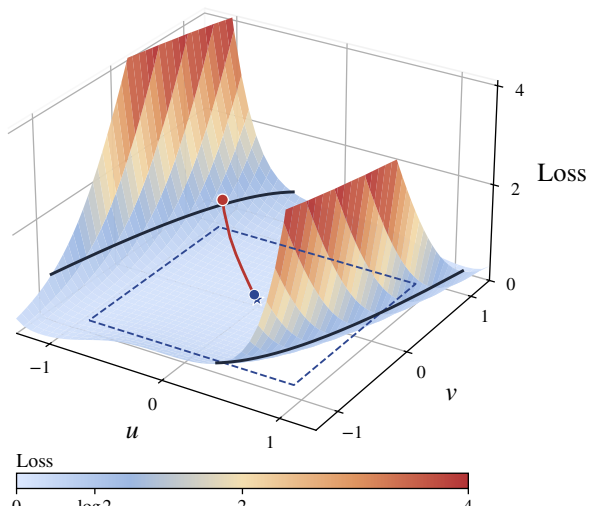
Loss = 0.13 margin = +1.998

Still correct

THE ATTACK MOVES THE INPUT UPHILL IN LOSS

Increase the loss to change the prediction

L-infinity steps use the gradient signs; the model stays fixed.



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Climb the same loss surface

Take small signed-gradient steps.
Same start as the FGSM example.
Here, the attack reaches its corner.

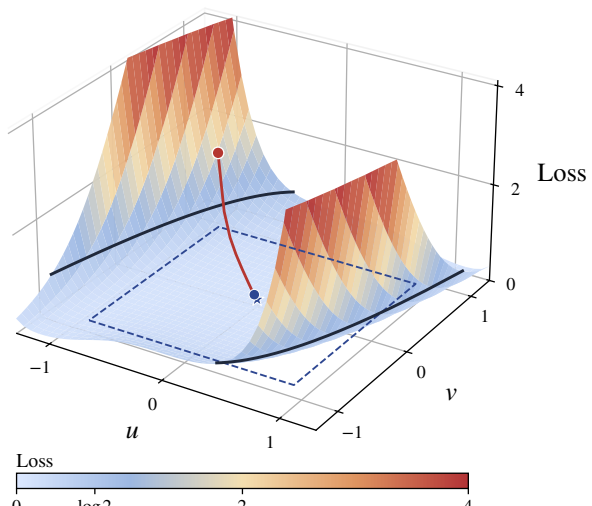
Loss = 0.74 margin = -0.088

Misclassified

THE ATTACK MOVES THE INPUT UPHILL IN LOSS

Increase the loss to change the prediction

L-infinity steps use the gradient signs; the model stays fixed.



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Climb the same loss surface

Take small signed-gradient steps.
Same start as the FGSM example.
Here, the attack reaches its corner.

Loss = 1.56 margin = -1.321

Misclassified

FGSM: A FIRST-ORDER ATTACK

For an ℓ_∞ budget, linearise the loss around x:

$$\ell(f_\theta(x+\delta), y) \approx \ell(f_\theta(x), y) + \delta^\top \nabla_x \ell(f_\theta(x), y).$$

FGSM: A FIRST-ORDER ATTACK

For an ℓ_∞ budget, linearise the loss around x:

$$\ell(f_\theta(x+\delta), y) \approx \ell(f_\theta(x), y) + \delta^\top \nabla_x \ell(f_\theta(x), y).$$

One step in the adversarial direction

$$x_{\text{adv}} = \text{clip}_{[0,1]}\big(x + \varepsilon\ \text{sign}\big(\nabla_x \ell(f_\theta(x), y)\big)\big)\,.$$

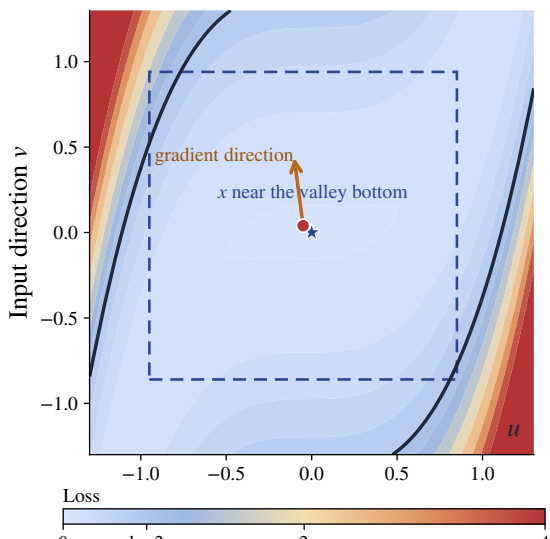
- Use the **sign**, not a normalised gradient: each nonzero component moves by $\pm\varepsilon$.
- With nonzero gradient components, this reaches a corner before pixel clipping.
- Fast: one gradient computation.
- Useful baseline; a one-step attack may miss adversarial examples.

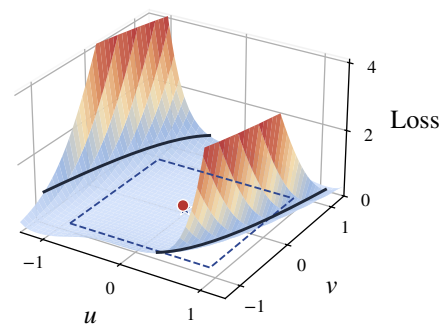
Goodfellow, Shlens & Szegedy · ICLR 2015

FGSM: ONE SIGNED STEP TO A CORNER

FGSM: one gradient, one jump

Read the gradient at the clean input.





$\delta_{FGSM} = \varepsilon \, \mathrm{sign}(\nabla_x \ell)$

Loss = 0.13 margin = +1.998

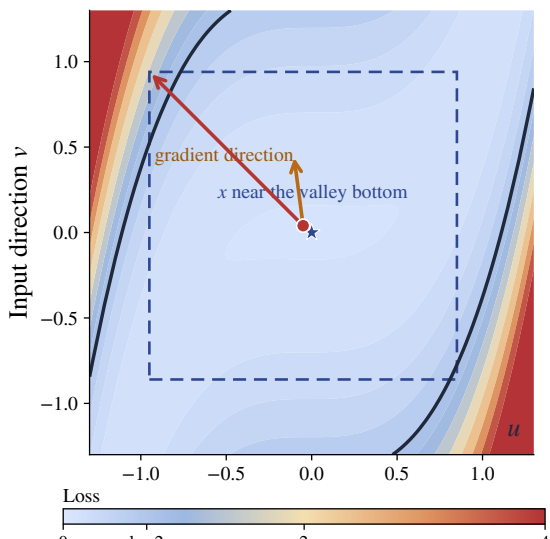
Still correctly classified

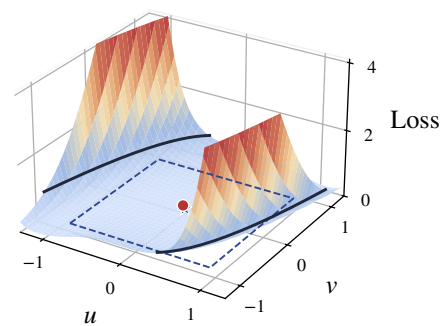
Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

FGSM: ONE SIGNED STEP TO A CORNER

FGSM: one gradient, one jump

Use its signs to choose one corner of the box.





$\delta_{FGSM} = \varepsilon \operatorname{sign}(\nabla_x \ell)$

Loss = 0.13 margin = +1.998

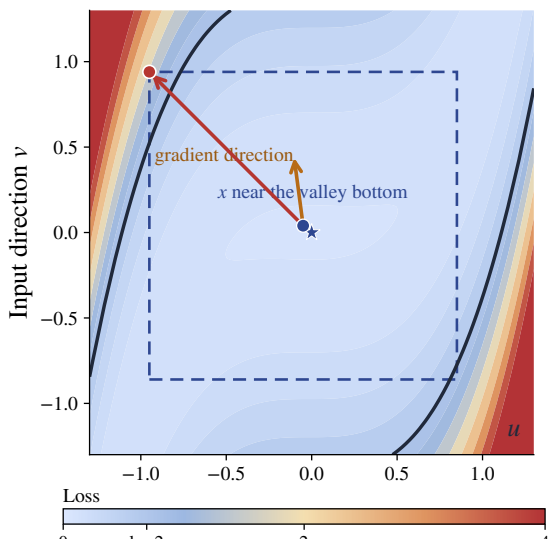
Still correctly classified

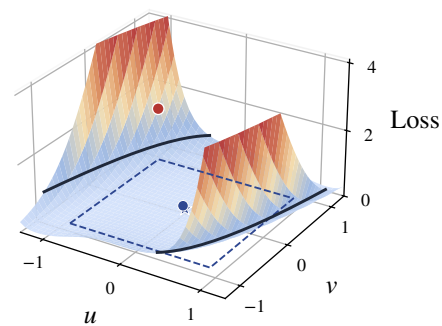
Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

FGSM: ONE SIGNED STEP TO A CORNER

FGSM: one gradient, one jump

One FGSM step: jump directly to that corner.





$\delta_{FGSM} = \varepsilon \, \mathrm{sign}(\nabla_x \ell)$

Loss = 1.56 margin = -1.321

Misclassified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

PGD: TAKE A GRADIENT STEP, THEN PROJECT

$$x_{t+1} = \Pi_{\mathcal{S}(x)}(x_t + \alpha \ \mathrm{sign}(\nabla_{x_t} \ell(f_\theta(x_t), y))) \,, \qquad \mathcal{S}(x) = \{u : \|u - x\|_\infty \leq \varepsilon\} \cap [0,1]^d.$$

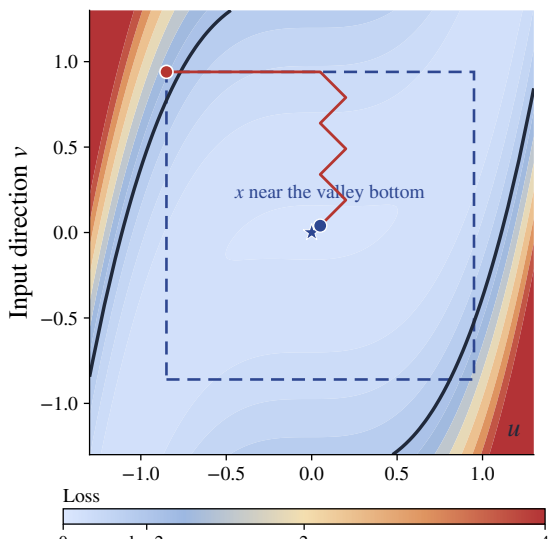
- ε: perturbation budget; α: step size; K: number of steps.
- A proposed step may leave the allowed set. Projection brings it back.
- Random initialisation and restarts can strengthen the search.

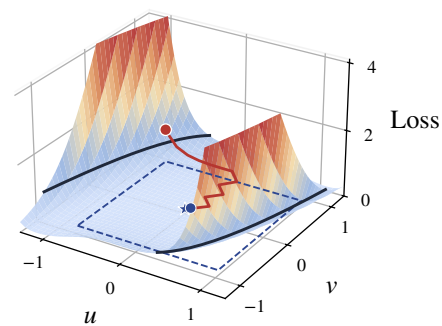
Madry et al. · ICLR 2018

PGD: FOLLOW THE POINT THROUGH THE ITERATIONS

PGD: step outside, then project back

Hold the point at the edge of the allowed box.





The current iterate is on the boundary

Loss = 0.97 margin = -0.488

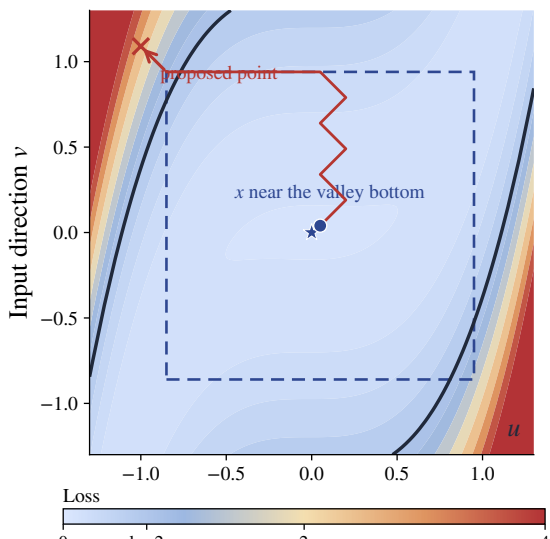
Misclassified

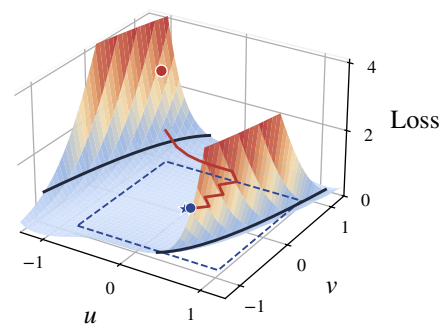
Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

PGD: FOLLOW THE POINT THROUGH THE ITERATIONS

PGD: step outside, then project back

This is a temporary proposal, not an admissible iterate.





The next proposed point is outside

Loss = 2.55 margin = -2.468

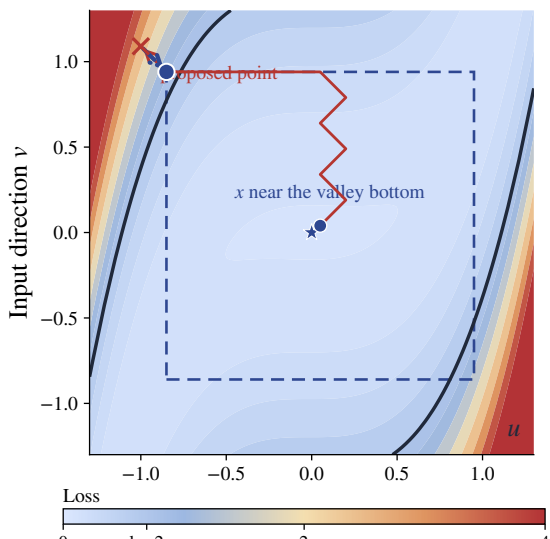
Misclassified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

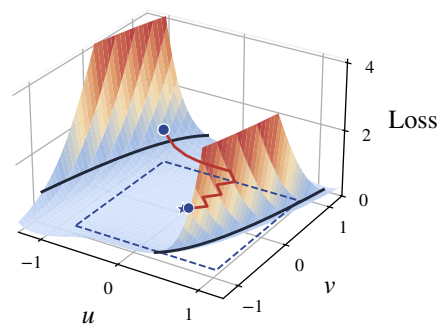
PGD: FOLLOW THE POINT THROUGH THE ITERATIONS

PGD: step outside, then project back

Projection returns the proposal to the boundary of the allowed box.



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.



Project back and keep the blue point

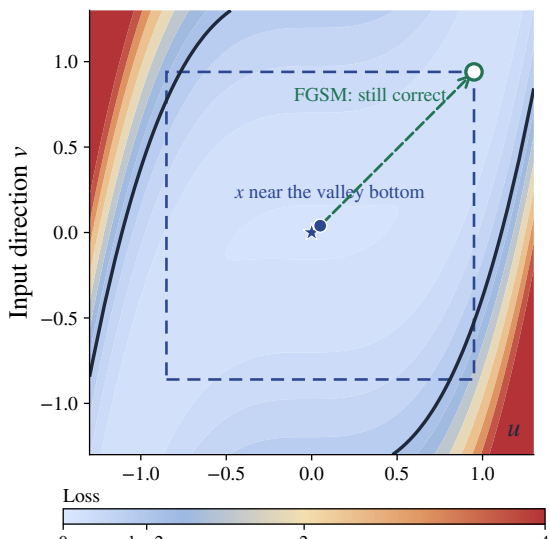
Loss = 0.97 margin = -0.488

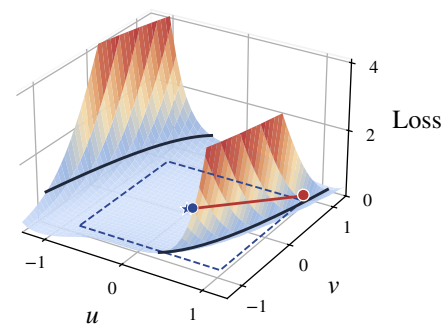
Misclassified

Same budget: FGSM can miss what PGD finds

One step can fail; several steps can succeed

FGSM jumps along the initial gradient signs, but stays correct.





Same model and budget; a nearby clean input

Loss = 0.14 margin = +1.903

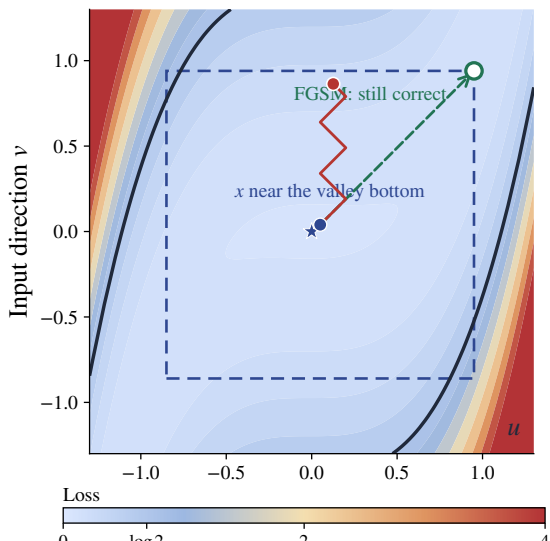
Still correctly classified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

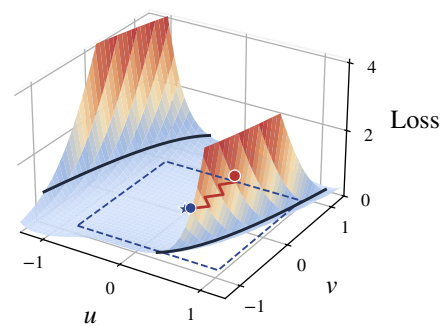
Same budget: FGSM can miss what PGD finds

One step can fail; several steps can succeed

PGD restarts at the same clean input, then recomputes the gradient at each step.



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.



Same model and budget; a nearby clean input

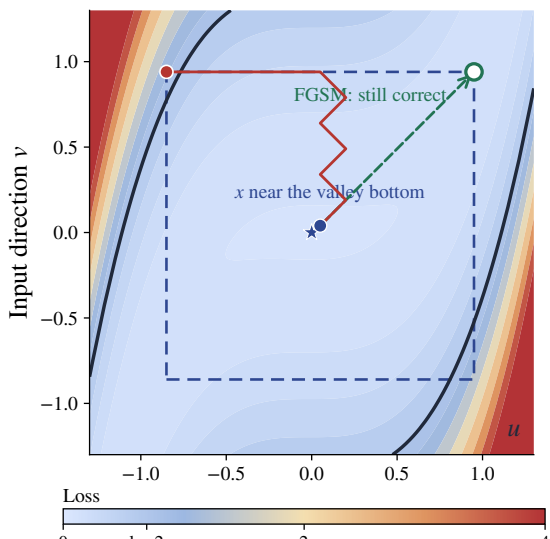
Loss = 0.25 margin = +1.256

Still correctly classified

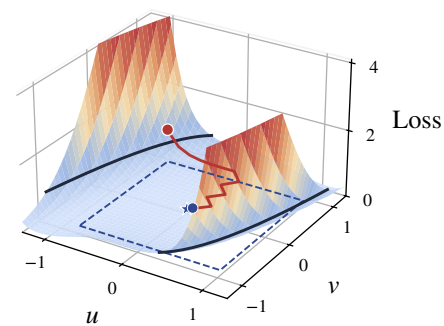
Same budget: FGSM can miss what PGD finds

One step can fail; several steps can succeed

PGD restarts at the same clean input, then recomputes the gradient at each step.



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.



Same model and budget; a nearby clean input

Loss = 0.97 margin = -0.488

Misclassified

Accuracy: three different statements

Clean accuracy

How many evaluation images are classified correctly **without perturbation**?

Accuracy under a specified attack

How many remain correct after **this attack**, with its norm, budget, steps and restarts?

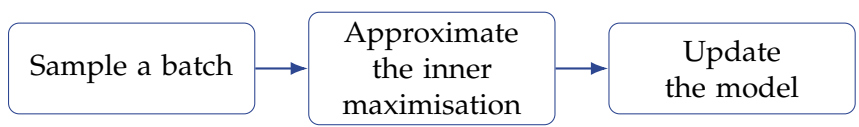
Worst-case robust accuracy

How many are correct for **every admissible perturbation**? A heuristic attack usually cannot establish this exactly.

Train a robust model

Adversarial training: optimise against a local adversary

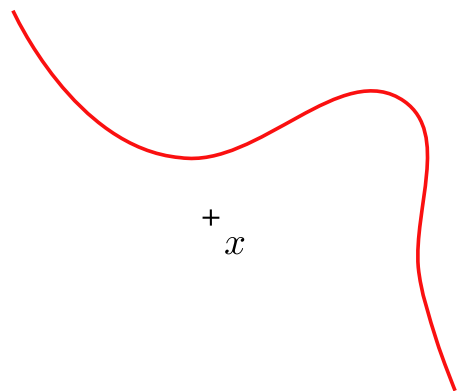
$$\min_{\theta} \mathbb{E}_{(x,y)} \left[\max_{\substack{\|\delta\|_p \leq \varepsilon \\ x+\delta \in [0,1]^d}} \ell(f_\theta(x+\delta), y) \right].$$



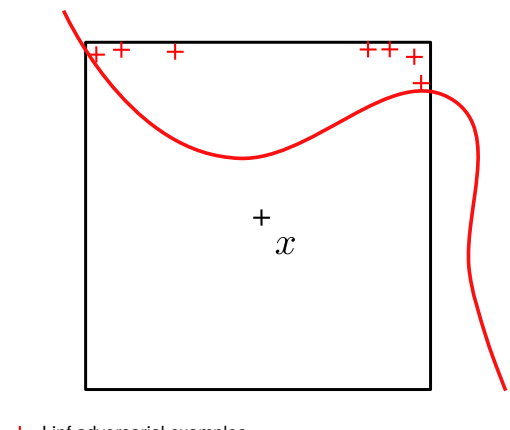
- FGSM or PGD approximates the inner problem: attack quality matters.
- Test attacks beyond the exact training configuration.
- Empirical robustness is not a certificate for every perturbation.

Goodfellow et al. · ICLR 2015; Madry et al. · ICLR 2018

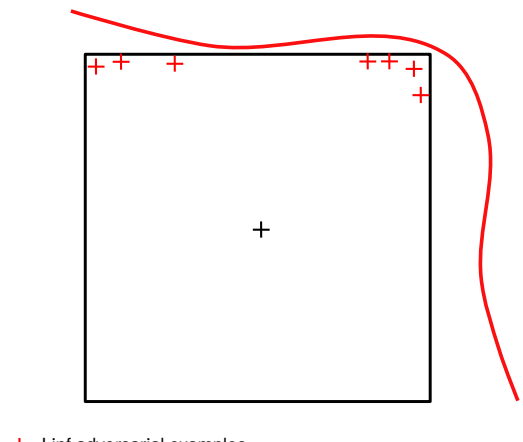
Robustness to one norm is not robustness to every norm



Robustness to one norm is not robustness to every norm

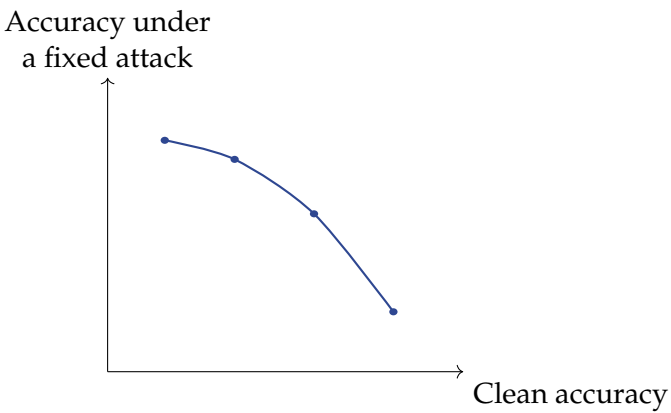


Robustness to one norm is not robustness to every norm



Training encourages stability inside the chosen neighbourhood.
Evaluate ℓ_∞ and ℓ_2 attacks separately; do not assume invariance.

Clean accuracy and robustness are both part of the result



Conceptual illustration, not measured data

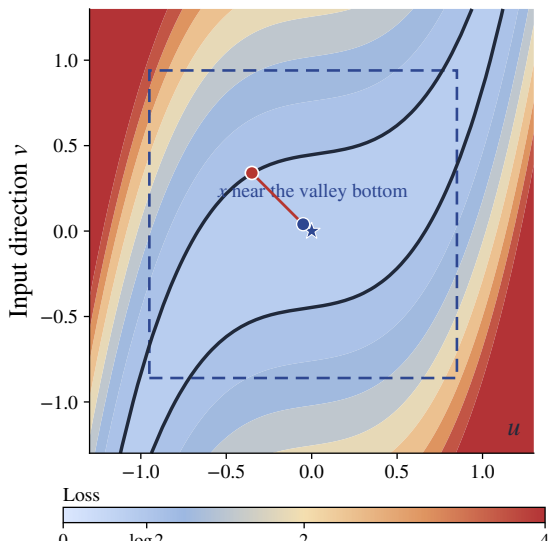
Zhang et al. · ICML 2019 TRADES

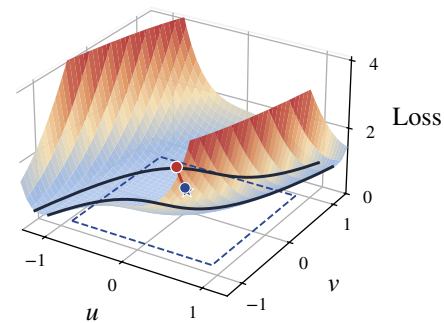
- Report both quantities using the same protocol.
- Vary one parameter and interpret the trend.
- Include training and inference cost in comparisons.

A gain under one attack can hide losses elsewhere.

Adversarial training: small, alternating updates

Round 1/5 — early-stopped PGD for this illustration





1. Stop just after crossing the boundary

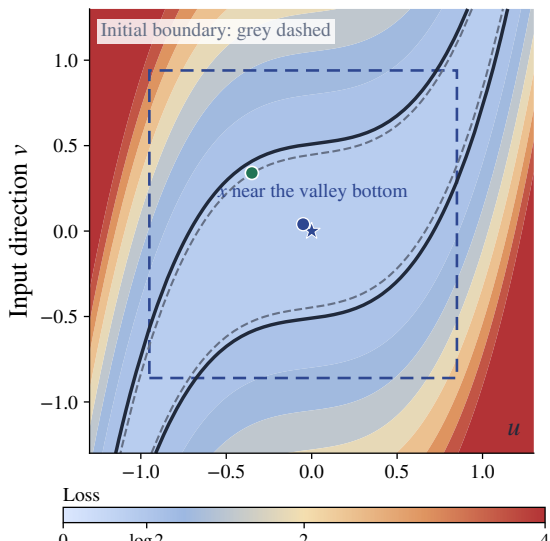
Loss = 0.694 margin = -0.002

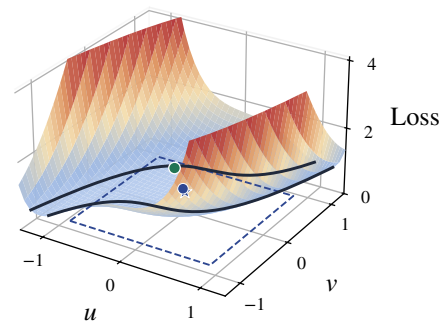
Misclassified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 1/5 — hold the point fixed; make one small SGD update





2. Push the nearby boundary back slightly

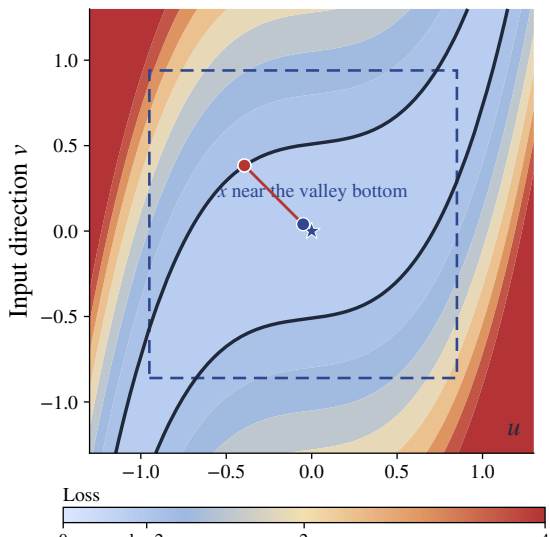
Loss = 0.662 margin = +0.064

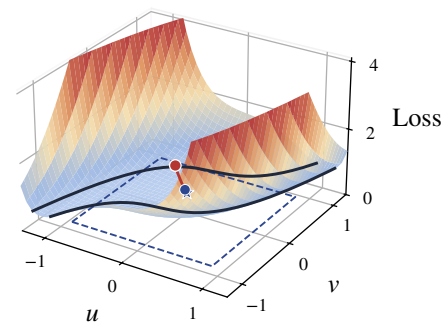
Still correctly classified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 2/5 — early-stopped PGD for this illustration





1. Stop just after crossing the boundary

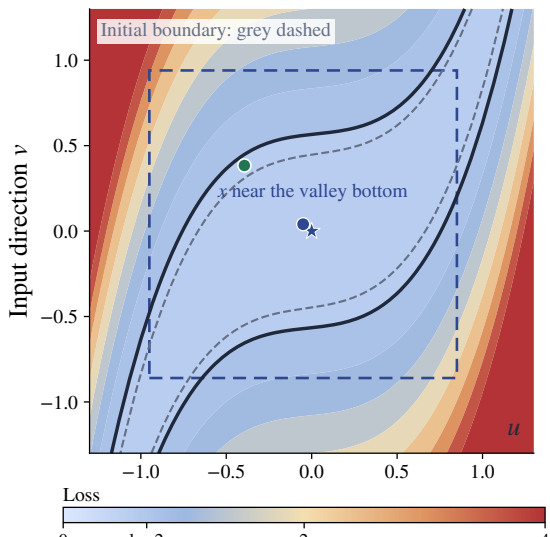
Loss = 0.694 margin = -0.002

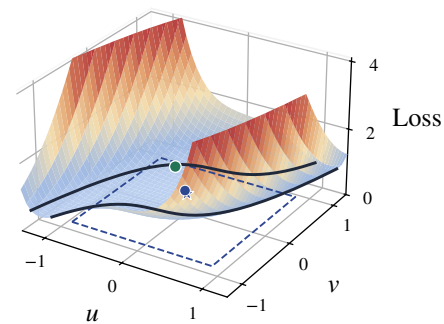
Misclassified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 2/5 — hold the point fixed; make one small SGD update





2. Push the nearby boundary back slightly

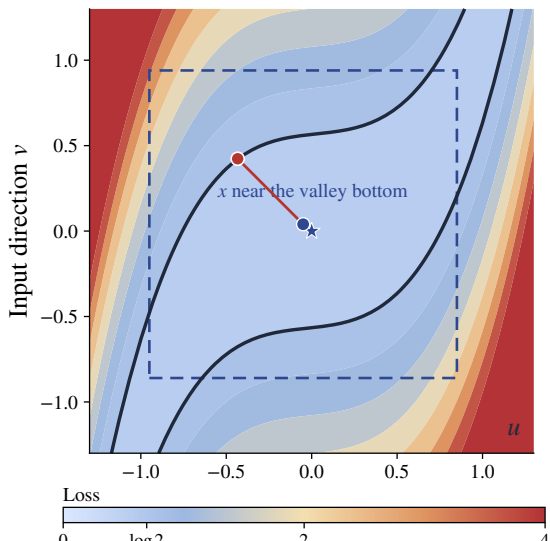
Loss = 0.660 margin = +0.067

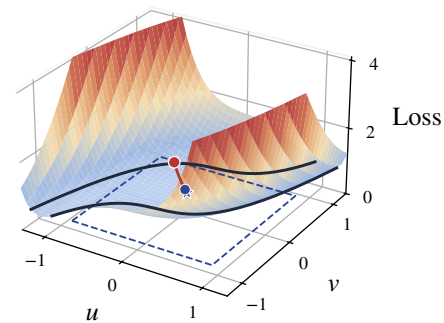
Still correctly classified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 3/5 — early-stopped PGD for this illustration





1. Stop just after crossing the boundary

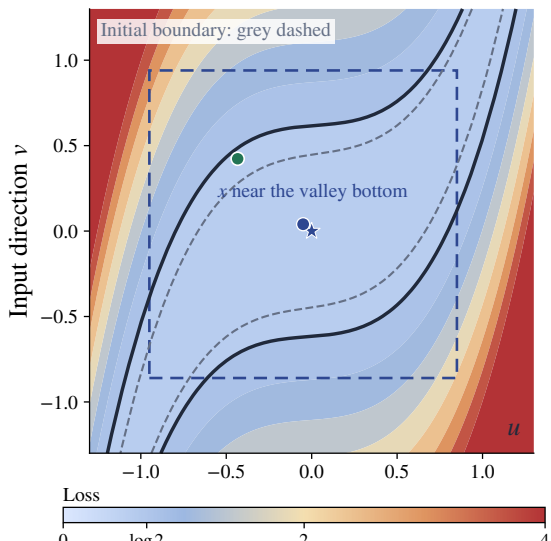
Loss = 0.694 margin = -0.002

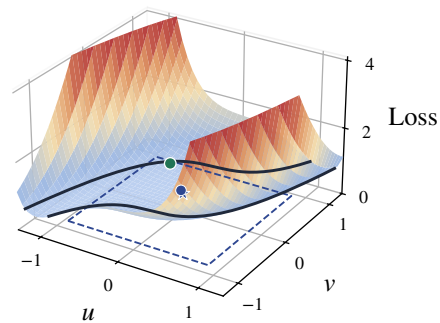
Misclassified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 3/5 — hold the point fixed; make one small SGD update





2. Push the nearby boundary back slightly

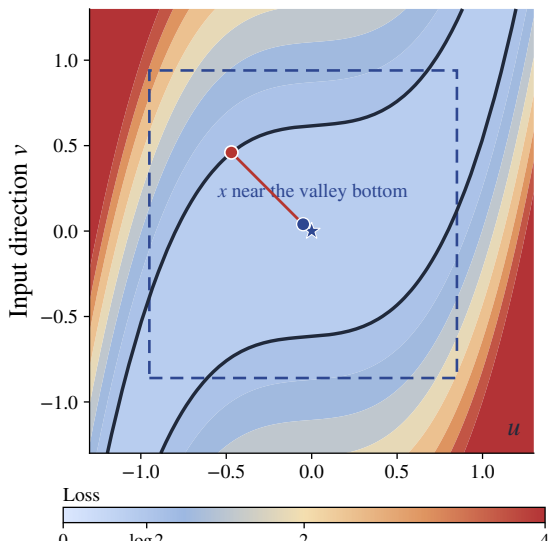
Loss = 0.658 margin = +0.072

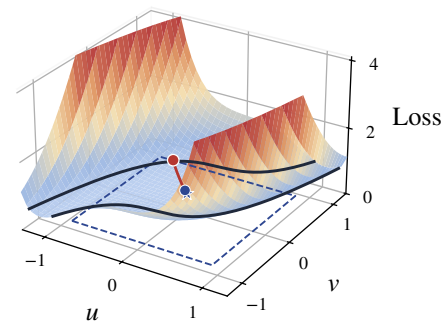
Still correctly classified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 4/5 — early-stopped PGD for this illustration





1. Stop just after crossing the boundary

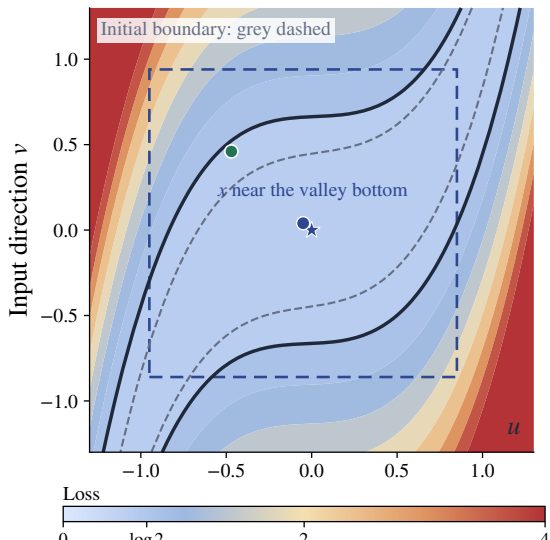
Loss = 0.694 margin = -0.002

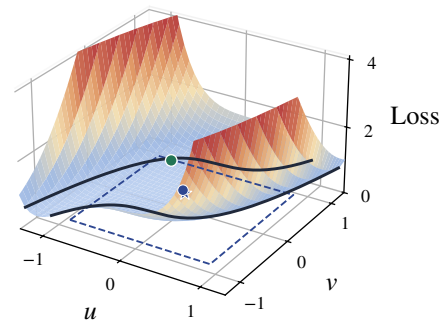
Misclassified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 4/5 — hold the point fixed; make one small SGD update





2. Push the nearby boundary back slightly

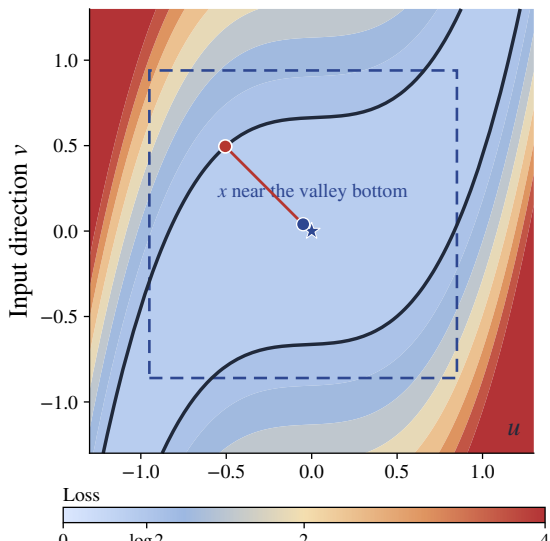
Loss = 0.655 margin = +0.078

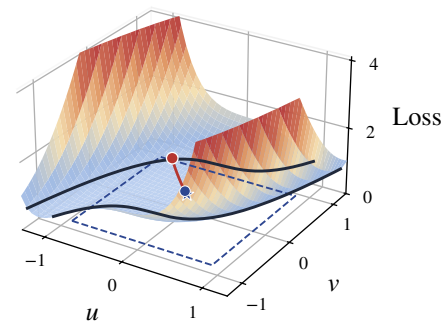
Still correctly classified

Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 5/5 — early-stopped PGD for this illustration





1. Stop just after crossing the boundary

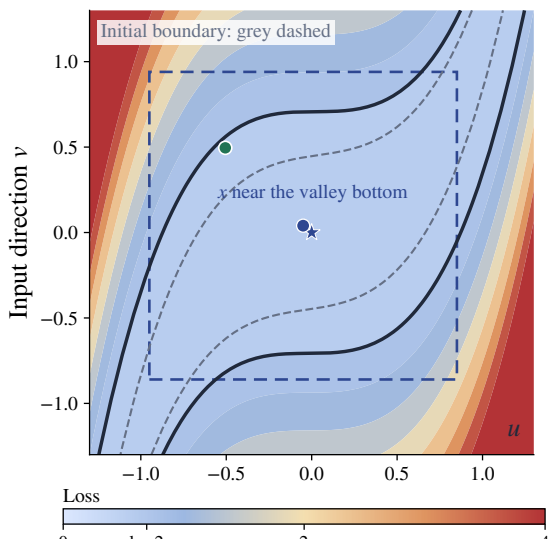
Loss = 0.694 margin = -0.002

Misclassified

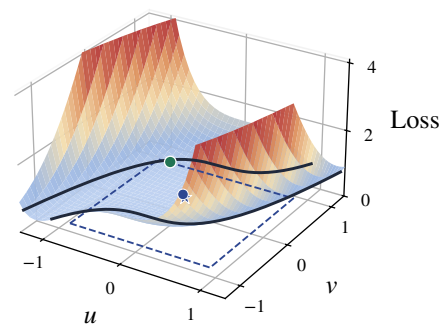
Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.

Adversarial training: small, alternating updates

Round 5/5 — hold the point fixed; make one small SGD update



Illustrative 2D binary classifier. Black curve: decision boundary; blue star: valley minimum.



2. Push the nearby boundary back slightly

Loss = 0.652 margin = +0.085

Still correctly classified

Randomisation: attack the model that is actually used

A stochastic model depends on a random variable ω: $f_\theta(x; \omega)$.

Estimate the gradient through randomness

$$\nabla_x \mathbb{E}_\omega[\ell(f_\theta(x;\omega), y)] \approx \frac{1}{M} \sum_{m=1}^{M} \nabla_x \ell(f_\theta(x;\omega_m), y).$$

Expectation over transformation / randomness (EOT).

Athalye et al. · ICML 2018: EOT; obfuscated gradients

- Different outputs alone do not demonstrate robustness.
- Attack the chosen prediction rule.
- Check the number of random samples.

Other defence mechanisms

Regularisation

Control sensitivity, constrain Lipschitz behaviour, or change the training objective.

Multiple threat models

Does an improvement transfer from one perturbation set to another?

Cohen, Rosenfeld & Kolter · ICML 2019 Randomized Smoothing

Randomised smoothing

A Gaussian-smoothed classifier can admit ℓ_2 certificates under a specific procedure and assumptions.

A noisy network alone is not a certified defence. Distinguish measured and certified accuracy.

Assignment 3

Start assignment A3 on Classroom 50

1. Open the A3 invitation and accept the assignment.
2. Form a group of **three**: one student creates the group, the other two join it.
3. Use the repository created for your group and verify that all three accounts are members.

▸ Accept A3 on Classroom 50

Course website: resources, deadlines and updates

From a baseline to a research question

First: a trustworthy pipeline

- CIFAR-10, a classifier and saved weights.
- Clean evaluation, then FGSM and PGD.
- An adversarial-training baseline.
- Verify splits, ranges and attack constraints.

Then: investigate one idea in depth

- A defence, objective or attack mechanism.
- A hypothesis and a controlled comparison.
- Ablations testing your explanation.
- Analysis of limits and computational cost.

The number of methods is not the objective. Make the evidence convincing.

Your model must be loadable independently

```python
from model import Net

net = Net()
net.load_for_testing(
    project_dir=project_dir
)
net.eval()
output = net(images)
```

- Entry: `model.py`, class `Net`.
- Input: `N x 3 x 32 x 32`, pixel values in $[0, 1]$.
- Baseline output: ten log-probabilities per image.
- Resolve weights relative to `project_dir`.

Do not train, download data or assume a specific working directory at import/load time.

How will the model be evaluated?

Clean and attacked CIFAR-10 images

- Clean accuracy reported separately.
- ℓ_∞ attack: $\varepsilon = 0.03$.
- ℓ_2 attack: $\varepsilon = 0.5$.
- Budgets apply to pixels in $[0, 1]$.

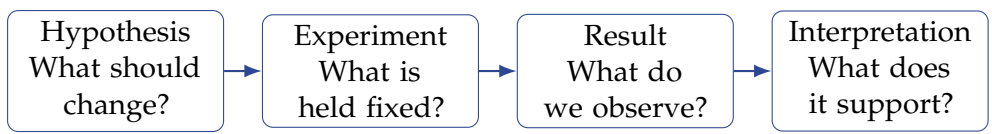
APGD-CE, via AutoAttack

- 100 iterations, one restart.
- EOT: five gradient samples.
- Predictions evaluated over five draws.
- Report the two attacked accuracies.

Leaderboard score: sum of the two attacked accuracies (0–200), not their average.

APGD-CE is the selected AutoAttack component, not the full ensemble. Stochastic accuracy averages correctness across draws, not majority voting.

An experiment should answer a question



- Same data split, threat model and attack configuration.
- One factor at a time; quantify uncertainty where feasible.
- For a parameter search, explain the trend, not just the winning value.
- Distinguish a plausible explanation from a supported conclusion.

A3 report: a small research paper

- **Question and method:** hypothesis and explanation of the mechanism.
- **Protocol:** enough detail to understand and reproduce the comparison.
- **Evidence:** readable figures, labelled axes/units and relevant ablations.
- **Analysis:** trends, unsuccessful cases, uncertainty and limitations.
- **Focus:** skip the textbook introduction; the reader knows the context.

Maximum five pages of scientific content, plus a group cover page. Optional appendices are not read.

Final presentations: four minutes per group

Thursday 3 December

14:15–17:30

Groups of **three students**.

Four-minute presentation.
An alarm rings at four minutes.

All groups give one final presentation; no preliminary session.

A focused scientific story

1. Question and method's intuition.
2. Decisive experiment and its result.
3. Interpretation, limitation and conclusion.

Use three or four readable slides. Rehearse the timing.

Questions, transitions and a break share the remaining session time.

Deadlines and responsibilities

Deadline (Paris time)	What to submit
2 December, 23:59	`slides.pdf` and `report.pdf` in your repository.
3 December, 05:00	Final code and checkpoint version for the last evaluation.
3 December, 14:15	Be ready for the final presentation session.

- AI allowed: understand, verify and defend the work.
- Code itself is not graded; scientific reasoning and communication matter.
- Do not make FGSM/PGD background the whole presentation.

Choose a direction, then investigate it properly

Direction	Starting points
Adversarial training	Goodfellow et al. (2015); Madry et al. (ICLR 2018).
Training objectives	Zhang et al. (ICML 2019): TRADES.
Randomisation / certification	Cohen et al. (ICML 2019); certification requires a specific procedure.
Reliable evaluation	Athalye et al. (ICML 2018); Croce & Hein (ICML 2020): AutoAttack.

Reading directions, not a requirement to implement every method.